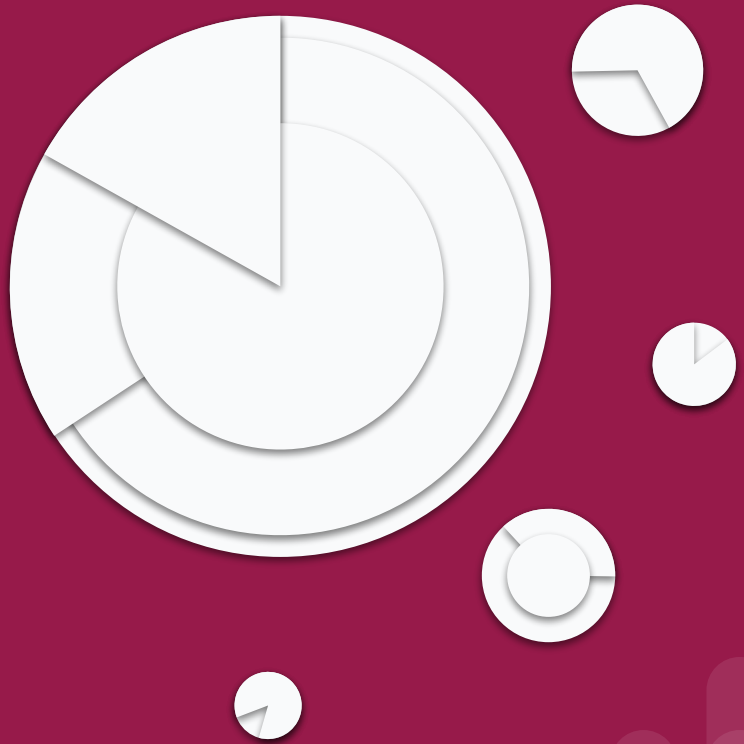
Data ScienceTech Institute

Clean IT

Hanna Abi Akl



Course Objectives

- Present common tools used by Data Scientists, Data Engineers and Data Analysts
- Familiarize students with basic installation and configuration
- Gain working understanding of different tools and their usage

Course Summary

- Python in Google Colab
- Python local setup (Anaconda)
- Python virtual environment
- Jupyter notebook overview
- IDE setup: Visual Studio Code
- R and RStudio local setup

Python in Google Colab

- Create a Google account here
- Access Google Colab here
- Google Colab formatting cheat sheet here
- As of April 2020, Colab runs Python 3.6
- Interactive workspace for code and text

Python in Google Colab

- The 3 environments used by Google Colab
  - Linux virtual machine hosted by Google
    - Lost when you leave or restart Colab
  - Google Drive
    - For storing your notebooks and/or datasets
  - Your local PC

Python in Google Colab

Data ScienceTech Institute

- Nice features of Colab
  - Enable GPU
  - Manage sessions
  - Easy to visualize work and try code without setup

Python in Google Colab

- Pros
  - Easy to launch
  - Free access to GPU
  - Collaborative
  - Free
- Cons
  - Limited use of GPU
  - Long sessions are killed
  - Data is lost if not saved on Drive/local disk

Python in Google Colab

- Try out Colab yourself
  - Import Colab_Exercise.ipynb in Colab
  - Try different methods to upload the colab_exercise_data.csv file (Local upload, Google Drive, ...)

Python local setup

- Download Anaconda here
- Version 3.8 is a good compromise
- Follow the installation wizard and check “Install for all users”
- Verify that you have both the navigator and command prompt installed

Data ScienceTech Institute

Python local setup

- **Check *Conda Cheat Sheet* for basic conda commands**
- ***conda --version* to check local conda version**
- ***python --version* to check local python version**
- ***conda list* to check installed packages**
- **Check available conda packages and installation here**

Python local setup (conda)

- Python libraries
  - Standard Library
    - Installed with Python
    - Modules are readily available
    - Found locally in Anaconda/Lib (or Anaconda3/Lib)
  - Packages
    - Can be installed in different ways (pip vs conda)
    - Found locally in Anaconda/Lib/site-packages

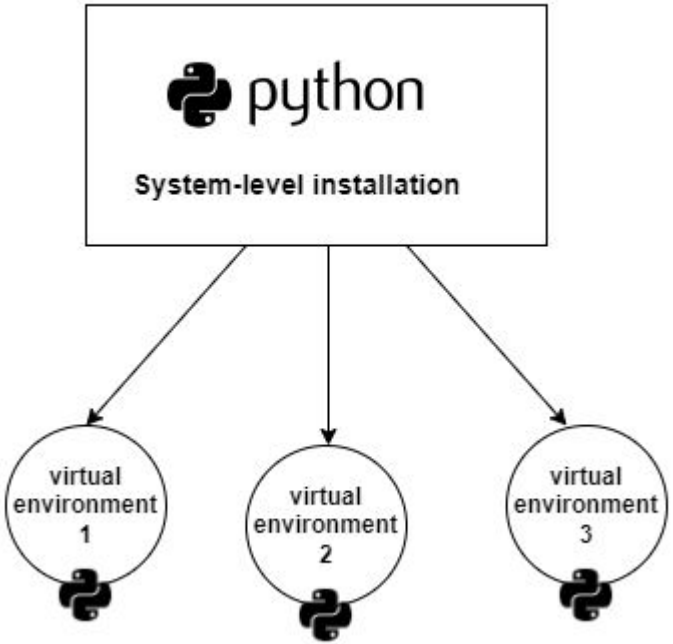
Python local setup

- **pip is already installed with Anaconda Windows version**
- *pip list* **to check all installed dependencies with pip**
- *pip freeze > requirements.txt* **to save environment current packages with pip**
- *conda list --export > requirements.txt* **to save environment current packages with conda**
- *pip install package_name* **to install a package with pip (e.g.** *pip install numpy***)**
- *conda install --channel "anaconda" package_name* **to install a package with conda (e.g.** *conda install --channel "anaconda" numpy***)**

Python virtual environment

- A virtual environment is a Python environment such that the Python interpreter, libraries and scripts installed into it are all isolated from those installed in other virtual environments and from the “system” Python

Python virtual environment



Python virtual environment

- Beneficial when working on different projects at the same time
- Some libraries may be incompatible
- We may want to use a different version of the library
- We may want to use a different Python interpreter
- Different tools exist to create a virtual environment (virtualenv, conda, ...)

Python virtual environment

- **Assign virtual environment to a project**
  - **Create the project directory**
  - *conda create - n envname* **to create virtual environment**
  - *conda activate envname* **to activate virtual environment**
  - *conda deactivate* **to deactivate virtual environment**
  - *conda env list* **to check all available virtual environments**

Jupyter notebook overview

- An “IDE” that makes it possible to execute code in blocks and add comments in a nice way
- Install it inside your virtual environment from the Anaconda Navigator
- Try opening the Colab_Exercise.ipynb notebook in Jupyter

IDE setup: Visual Studio Code

- IDE = Integrated Development Environment
- Helps write and manage code
- Different IDEs are available
- Popular Python IDEs: Pycharm, Visual Studio Code

IDE setup: Visual Studio Code

- Install Visual Studio Code here
- Create a project directory and from the folder use the command “code .” in your terminal to open the project in Visual Studio Code
- Install extension: Python microsoft (to manage python code)
- To change default shell: “CTRL+SHIFT+P” + “Terminal: select default profile”
- You can also change your interpreter: “CTRL+SHIFT+P”

IDE setup: Visual Studio Code

- You can activate a virtual environment inside Visual Studio Code
- Install Linter: Pylint (for tracking syntax errors)
- Try debugging your code in Visual Studio Code

Data ScienceTech Institute

R local setup

- Install R here
- Install RStudio here
- R is a programming language (like Python!)
- Rstudio is an IDE to write R code (like Visual Studio Code!)
- Similar to Python, install packages to be used
- Fun fact: R can also be run inside a Jupyter Notebook here